

07/18/2003 17:56 FAX

Attorney's Docket No.: 10454-016001 / P4361

Applicant : Lincoln et al.
Serial No. : 09/855,458
Filed : May 15, 2001
Page : 2 of 28

In the specification:

Please amend the paragraph beginning at page 1, line 4 as follows:

a¹ This application claims priority to U.S. Provisional Application No. 60/289,573 filed on May 10, 2001, entitled "Modeling Biological Systems", naming ~~Patrick D. Lincoln and Keith R. Laderoute as inventors~~, the contents of which are incorporated herein in their entirety by reference.

Please amend the paragraph beginning at page 5, line 4 as follows:

a² The method may also include outputting the terminal state or at least one of the alternative resultant states, e.g., as a graphical display. The method may also include outputting large sets of possible terminal states as a set of alternatives, e.g., as as a tree or graph representing those possible terminal states. The method may include navigation of such possible terminal states with logical operators.

Please amend the paragraph beginning at page 6, line 21 as follows:

a³ The methods described here generally use the semantic and logical framework of formal methods to model the circuitry of biological systems. Referring to the example shown in Fig. 1, information about a biological system is extracted 110, e.g., from direct observations, experiments, and scientific literature (e.g., PubMed at the National Center for Biotechnology Information, National Institutes of Health, <http://www.ncbi.nlm.nih.gov/entrez/>). Elements in the biological system are identified and represented 120 as symbols. Symbols can be related to one another using hierarchical data types. The information is then parsed to formulate 130 rules about the system. The symbols and rules can be used in a variety of methods. Non-limiting examples include evaluating a hypothetical initial state for the biological system 140 and 142; testing a theorem 150 and 152; and checking a model.

Please amend the paragraph beginning at page 8, line 6 as follows:

a⁴ Rewriting change can occur independently from any other non-overlapping state change. Hence, the rewriting rules can evaluate concurrent state changes, e.g., for highly

Applicant : Lincoln et al.
 Serial No. : 09/855,458
 Filed : May 15, 2001
 Page : 3 of 28

Attorney's Docket No.: 10454-016001 / P4361

nondeterministic concurrent computations. When applied to a hypothetical state of a system, a set of "terminating" rules can arrive at a solution for which no further state transitions can be applied. For example, the set of rules:

$$a:b \rightarrow b:a, b:a \rightarrow c$$

is terminating. When the input state is "ab," the rewriting rules reach the solution, "c". In contrast, "non-terminating" rules do not reach such a solution. For example, the set of rules:

$$a:b \rightarrow b:a, b:a \rightarrow a:b$$

is non-terminating. When the input state is "ab," the rewriting rules do not reach a solution for evaluating the system. ~~Non-termination~~ Non-terminating rules can cause "infinite substitution chains." The inference engine can be configured to detect such infinite substitution chains. Further, the two nonterminating rules above effectively express commutativity. An inference engine may detect rules expressing commutativity of an operator, and replace them with the explicit notation that the operator is commutative. This requires the inference engine to have built-in treatment of commutativity, e.g., including commutative matching. Similarly, the inference engine can detect rules expressing associativity. Such features of the inference engine are termed associative-commutative matching.

Please amend the paragraph beginning at page 10, line 14 as follows:

Two examples of computer environments that support rewriting logic are (1) the PVS specification language and theorem prover, and (2) the Maude rewriting engine. Both are available from the SRI Computer Science Laboratory (SRI International, ~~Palo Alto~~ Menlo Park CA; <http://www.csl.sri.com/>). Other examples of computer languages useful for formal methods include Café (Futatsugi and Sawada (1994) "Café as an Extensible Specification Environment" In *Proc. Kunming International CASE Symposium*) and ELAN (Borovansky *et al.* (1996) "Controlling Rewriting by Rewriting" in *Proc. First International Workshop on Rewriting Logic*, Electronic Notes in Theoretical Computer Science. Vol. 4 Elsevier). For a review of rewriting logic see, e.g., Meseguer (1998) "Research Directions in Rewriting Logic" In *Computational Logic*, ed., Berger and Schwichtenberg, Springer-Verlag; Meseguer (1996) *Proc. First*

07/18/2003 17:57 FAX

Attorney's Docket No.: 10454-016001 / P4361

Applicant : Lincoln et al.
 Serial No. : 09/855,458
 Filed : May 15, 2001
 Page : 4 of 28

International Workshop on Rewriting Logic, Electronic Notes in Theoretical Computer Science.
 Vol. 4 Elsevier; and Kirchner and Kirchner (1998) *Proc. Second International Workshop on
 Rewriting Logic*, Electronic Notes in Theoretical Computer Science. Vol. 15 Elsevier.

Please amend the paragraph beginning at page 10, line 29 as follows:

Maude is a computer-based language that efficiently supports rewriting logic computation, equational computation, and algebraic specification. Some algebraic features of Maude are implemented in the OBJ style (Goguen *et al.* (2000) "Introducing OBJ" In *Software Engineering with OBJ: Algebraic Specification in Action*, pp. 3-167, Kluwer) . Maude rewriting logic uses ~~algebraic~~ algebraic determination to identify all possible configurations of a system, e.g., a system subject to concurrent changes. With standard hardware, such as a Pentium II processor, the raw rewriting speed of Maude is over 10 million rewrites per second for simple rule sets.

Please amend the paragraph beginning at page 23, line 13 as follows:

Protein-protein interactions are common features of a biological system and can be central to generating a network of interactions, particularly in signaling pathways. Matrices of protein-protein interactions are available such as described in Walhout *et al.*, *Science* 287: 116-122, 2000; Uetz *et al.*, *Nature* 403, 623-631, 2000; and Schwikowski (2000) *Nature Biotech.* 18:1257. Walhout *et al.* identified interactions among a matrix of *C. elegans* ~~vulval~~ vulval development signalling proteins. Uetz *et al.* and Schwikowski *et al.* identified interactions among a matrix of thousands of proteins identified in yeast. The two-hybrid assay is used to determine if one polypeptide can bind to another. Since the assay is performed in yeast cells using fusion proteins to a DNA binding domain and a transcriptional activator, each yeast strain bearing a DNA binding domain fusion can be combined with yeast strain bearing the activation domain fusion using a simple mating technique. The assay is also easily scored by assessing reporter gene transcription. Each observed protein interaction can be used to generate an interaction rule, or a testable theorem. For example, if protein A stably binds to protein B, a rewriting rule can be used to replace the instance of protein A and B with a protein complex A:B.

Applicant : Lincoln et al.
Serial No. : 09/855,458
Filed : May 15, 2001
Page : 5 of 28

Attorney's Docket No.: 10454-016001 / P4361

Please amend the paragraph beginning at page 26, line 12 as follows:

Other databases of metabolic and signaling pathways include the WIT server at the
Argonne National Laboratory (~~http://www-unix.mcs.anl.gov/compbio/~~), and KEGG (Ogata *et al.*
(1998) *BioSystems* 47:119; ~~http://www.genome.ad.jp/kegg/~~).

Please amend the paragraph beginning at page 26, line 12 as follows:

The Appendix of U.S. Provisional Application No. 60/289,973, filed May 10,
2001, titled "Modeling Biological Systems," naming Patrick D. Lincoln and Keith R. Laderoute
as inventors, provides an example of Maude code for modeling cellular behavior.
